White Paper

Penetration Testing as Code
Future of offensive Security

prancer

Cloud Security Validation at Scale

Authors

Farshid Mahdavipour
CEO, Prancer

Farshid, Founder and CEO at Prancer Enterprise, has more than 20 years of experience in Information Systems, with the last ten years focused on Cloud Technologies and Security.

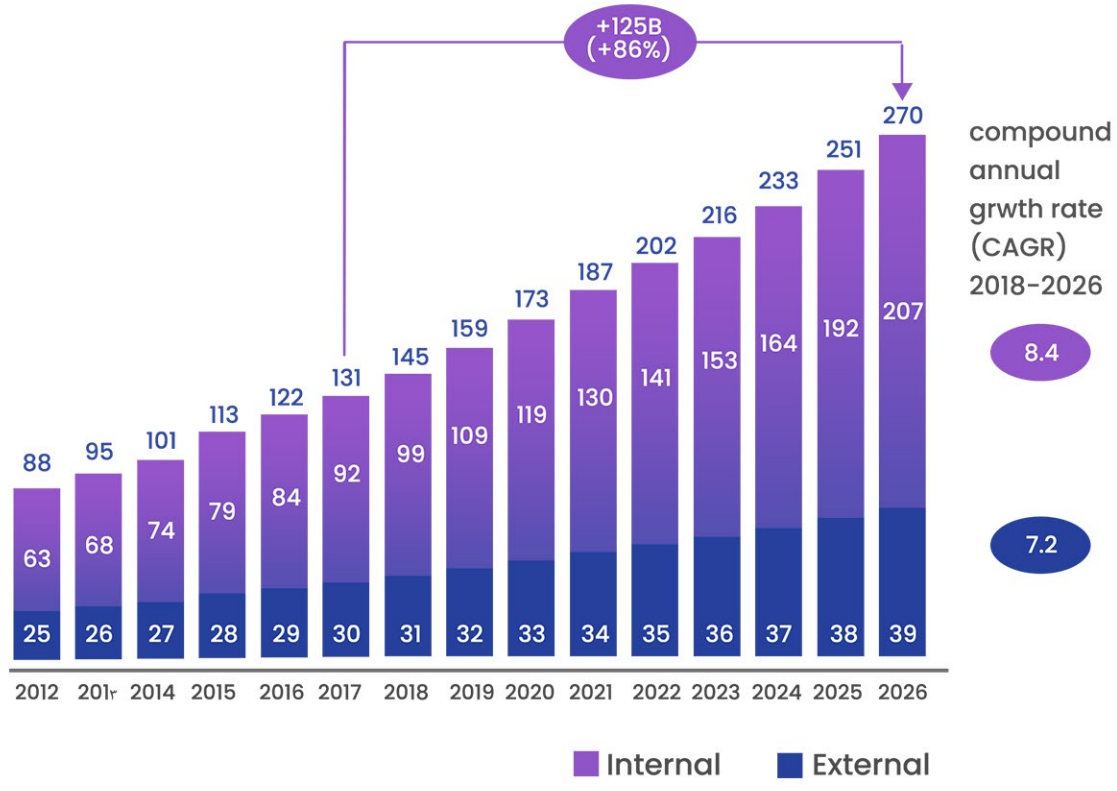
Kumar Chandramoulie
CPO, Prancer

Kumar, the chief product officer (CPO) at prancer, is a veteran cybersecurity executive with extensive experience operating large-scale offensive security operations for fortune top companies.

Introduction

In today's world, businesses need to have a plan in place when their systems are attacked, as cyber threats are getting persistent and advanced. Adversaries understand common defense techniques and deploy countermeasures to weaponize new and old vulnerabilities.

Cloud consumers employ a wide range of cloud security solutions to mitigate these threats. Most enterprises in the public cloud utilize a blend of 6-12 third-party security products or open-source toolings spread out across multiple domains, such as Static/dynamic scanners, security posture management, workload protection, endpoint protection, and security incident detection and response platforms.



Buying security is pointless without extensive testing

The proliferation of security tooling has tremendously increased end-to-end visibility, which has led to more cyber security findings at the expense of bigger cybersecurity budgets. Despite all of these defenses for a variety of threats and vulnerabilities, there is no assurance that your resources are truly secure end to end unless all defense-in-depth security controls are constantly tested across various attack vectors.

To stay secure, organizations need to adopt a “test your own defense” strategy. This should be a consistent tool in any organization’s arsenal to tackle the growing advanced cyber threats.

One way it has been performed is through the use of penetration testers, who are experts in finding and exploiting vulnerabilities in systems. However, this process can be time-consuming and expensive. The future of offensive security will be built on automation, not human intervention.

This whitepaper discusses the pain points around the effectiveness of today's cloud security validation and how Prancers Penetration Testing As Code (PAC) may help you build future-generation attack-ready clouds focused on offensive security that is both efficient and cost-effective.

01 Manual Pentesting (MPT) 1.0 - An overview

A vulnerability assessment is a process of identifying, quantifying, and ranking vulnerabilities in systems or networks. A penetration test is the simulated exploitation of those vulnerabilities to assess the damage that could be done.

Penetration testing is a critical part of compliance and regulation for several reasons. One, because it's the best way to identify and fix vulnerabilities before they can be exploited. Two, because it provides external assurance that your security controls are adequate. and most importantly it aids with white-box or grey box security testing of business logic in proprietary apps to verify data protection measures such as role-based data authorization and data exfiltration controls across a variety of conditions.

Pentesters are typically skilled in various techniques, and they can accurately diagnose an organization's state of vulnerability and then identify the best possible solutions to mitigate the risks. If you want to perform an internal audit or assess the cyber security risks within your business to assure you that everything is in working order, there's no better way than to pentest either manually or through simulated attacks.

02 Why manual pentests are setting organization for failure?

The biggest challenge with manual vulnerability assessments and penetration testing is that it's slow, expensive, and doesn't scale with modern CI/CD strategies. It can take weeks or even months to find and exploit all the vulnerabilities in a system. And it's challenging for security teams to keep up with the ever-changing landscape of security threats.

The other challenge is that most pentesters are skilled in various techniques to diagnose an organization's state of vulnerability accurately, but it also makes the process more expensive. You need to hire multiple experts who can cover a wide range of security disciplines such as microservices and APIs.

Finally, there's the human component. Pentesters are humans, and humans make mistakes. It's conceivable that they overlook critical flaws or miss important clues that might be used to exploit the system, and it isn't always feasible for red teamers to report risk meditations since they aren't always familiar with the systems or don't have the required knowledge and skills to fix the vulnerabilities they discover.

There is also a critical aspect about the timing of the findings, given all these pen tests are done after going live or just before going live which had proved to be a costly fix.

Manual pentesting is expensive for several reasons, including prep time, access and management overhead, infrastructure demands, testing cycles, triage time, communication and readout times, and the absence of historical reports.

Considering these facts, pentesting applications for a medium-sized company could cost up to hundreds of thousands of dollars per year. These challenges have led many organizations to consider automation to improve their penetration testing processes. Automation can help speed up the process, reduce costs, and minimize human errors.

03 Automate your penetration testing organically - Rescue mode

An automated penetration test, unlike manual testing, does not require the pentester to take any action. You must still write, build, execute, and test the code that performs such an evaluation but such actions would be taken before any human involvement is required. The result is faster scans than useless human resources.

The apparent benefit of automating the penetration testing process is that it saves time, reduces cost, and can be repeated continuously. Automated pentesting also enables better testing coverage since it requires the tester to only write test cases instead of performing all steps manually. This increases speed and accuracy even further during security assessments.

More important thing is that the developers can fix the code early in the cycle which saves developer fatigue and organizations cost-saving benefits inclusive of time to market, employee satisfaction, and customer confidence.

04 Context less DAST shifting to impactful correlation of findings

he DAST tooling may not be sufficient for security teams that support modern cloud-native applications, because it can't analyze certain contemporary web applications or APIs in a contextual white-box view for container and serverless apps where embedding agents on the runtime networks is no longer feasible. Adding be-spoke business logic to authenticated scans, on the other hand, is difficult in DAST evaluation. Moreover, removing false positives is not simple, and finally, DAST scan in the SDLC after CI/CD process jeopardizes developer productivity and shift-left ideology.

	Features	Manual Pentesting	DAST	PAC
Vulnerability Assessment for Cloud Applications		X	X	Y
	Cloud Infrastructure As Code (IAC) scanning	X	X	Y
	Automated cloud applications discovery	X	X	Y
	Cloud application scanning	X	Y	Y
	Internal scanning (white box)	Y	Y	Y
	External scanning (black box)	Y	Y	Y
	Detect 100+ web vulnerabilities including OWASP top 100	Y	X	Y
	API scanning	Y	X	Y
Peneteration testing capabilities	Real time Threat exploitation	Y	Y	Y
	Exploit lateral movements	Y	X	Y
	Authenticated scanning	Y	X	Y
	Token based authenticated scanning	X	X	Y
	Applications dashboard	X	X	Y
	Findings and documentation	Y	Y	Y
	Finding Triage and risk management	Y	Y	Y
	threat co-relation and risk contexts	X	X	Y
	Vulnerabilities As Code (VAC)	X	X	Y
	Remediation validation	X	X	Y
Automated Pentesting	Managed pentesting via code	X	X	Y
	Zero touch Managed Infrastructure (ZMI)	X	X	Y
	Serverless deployments	X	X	Y
	Container based scanner	X	X	Y
	YAML generators	X	X	Y
	Offline CI scanners	X	X	Y
	Real time CD scanners	X	X	Y
	CI/CD tooling integrations	X	X	Y
	Continous scanning	X	X	Y
	CLI support	X	X	Y
	Open API & swagger integration	X	X	Y
Enterprise Support	Professional Services	Y	Y	Y
	Dedicated Slack and Teams support	X	X	Y
	24*7 phone support	X	Y	Y
	JIRA, Slack integration	X	Y	Y
	Hosted versions	X	Y	Y

05 Automated pentesting - a perfect fit for cloud applications at scale

Automated pentesting is an excellent fit for modern cloud applications for a variety of reasons. One of the primary advantages of automated pentesting is that it may be expanded to meet the demands of agile cloud applications. Automated pentesting may be run in parallel on a large number of systems, and it can be integrated into the CI/CD process to guarantee that security is built right into the application from the start. This makes it an ideal match for cloud apps that are constantly being scaled up and down to meet the changing demands.

06 Security Validation as a code - The future

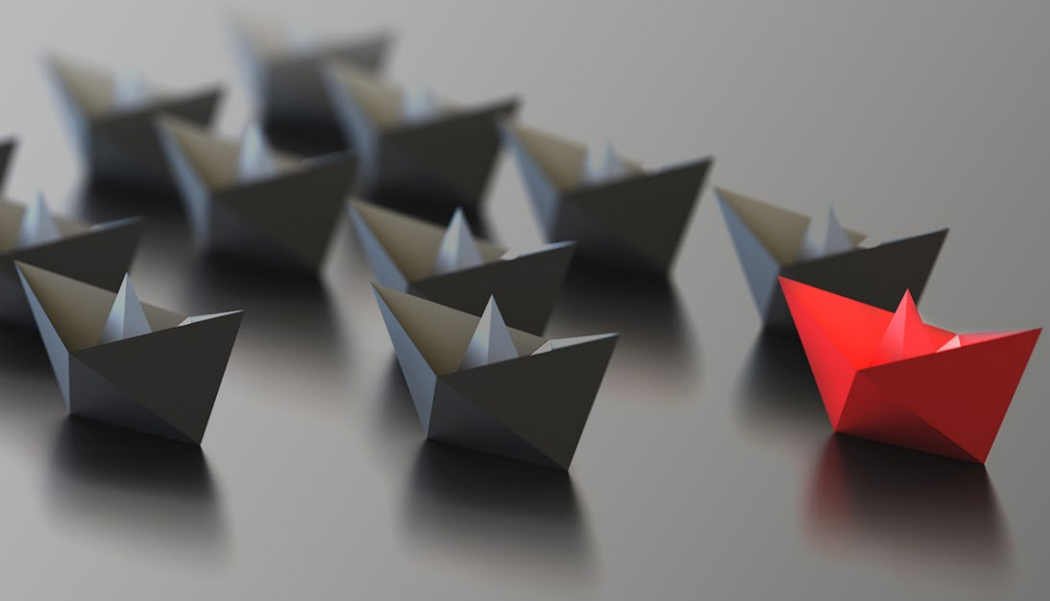
The majority of the time, security validation is a manual operation. It lacks the repeatability and process hygiene associated with SDLC. In the CI/CD world, the existence of a manual security testing procedure creates significant operational inefficiencies. Validation as code strives to minimize these barriers. For one, it helps to speed up the procedure by automating many of the operations that are currently done manually to co-exist with SDLC processes. It guarantees repeatability, accuracy, and consistency by removing human error.

Our goal with PAC is

to make offensive security tools accessible to product development teams. Traditional methods demand a significant amount of work from security experts and pentesters, who must manually repeat procedures that lack the reproducibility and process hygiene of software development processes. In today's CI/CD world, the existence of a manual security testing procedure creates significant operational inefficiencies. PAC strives to minimize these barriers".

said Prancer CEO & Founder Farshid Mahdavipour.

"Instead, the deep testing only happening at the end of a project or after a feature is built, which could be very costly to fix the code after the release of the product"



07 Offensive security at scale and with ease

The process of building an automation suite is often easier than hiring multiple experts for manual pentesting. Various tools are also available that can automate most of the tasks required for pentesting, including security scanning against cloud architectures that are built around microservices and APIs. This ability to automate manually intensive tasks means that organizations can speed up their validation process.

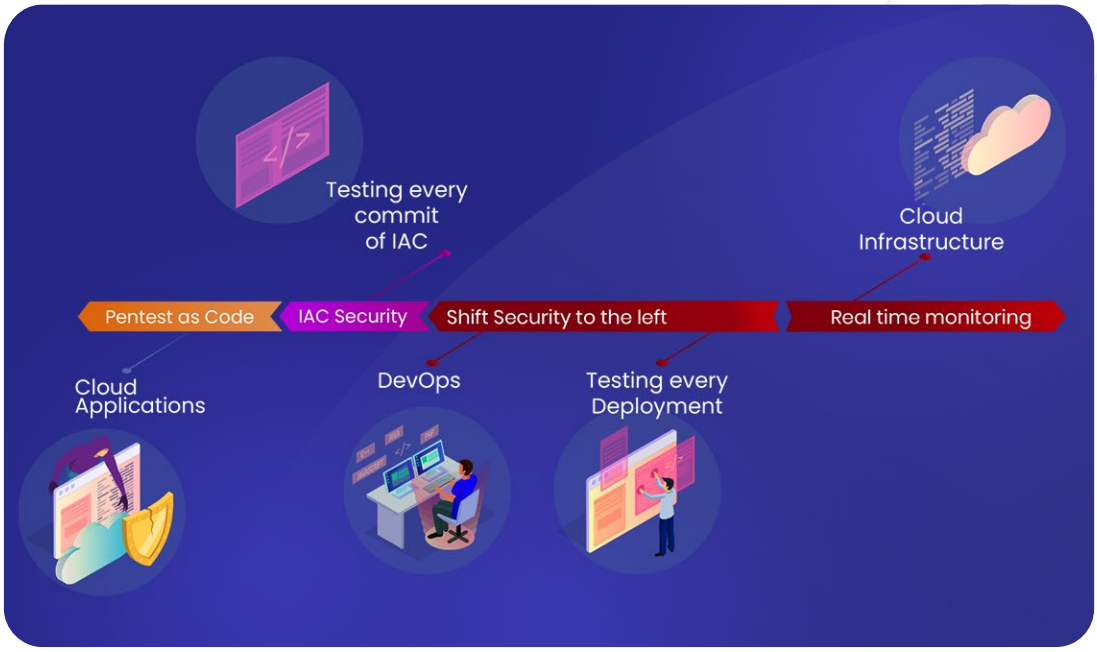
Prancer's Penetration Testing As Code Framework (PAC) provides "pentest as a service" by automating the process of scaling and deploying any number of serverless pentest instances on all major cloud providers.

08 Build attack ready cloud with shift-left

As large-scale cyber attacks continue to appear on a daily basis, it's critical for CISOs to protect their businesses against vulnerabilities before the exploits and build an attack-ready cloud from the inception.

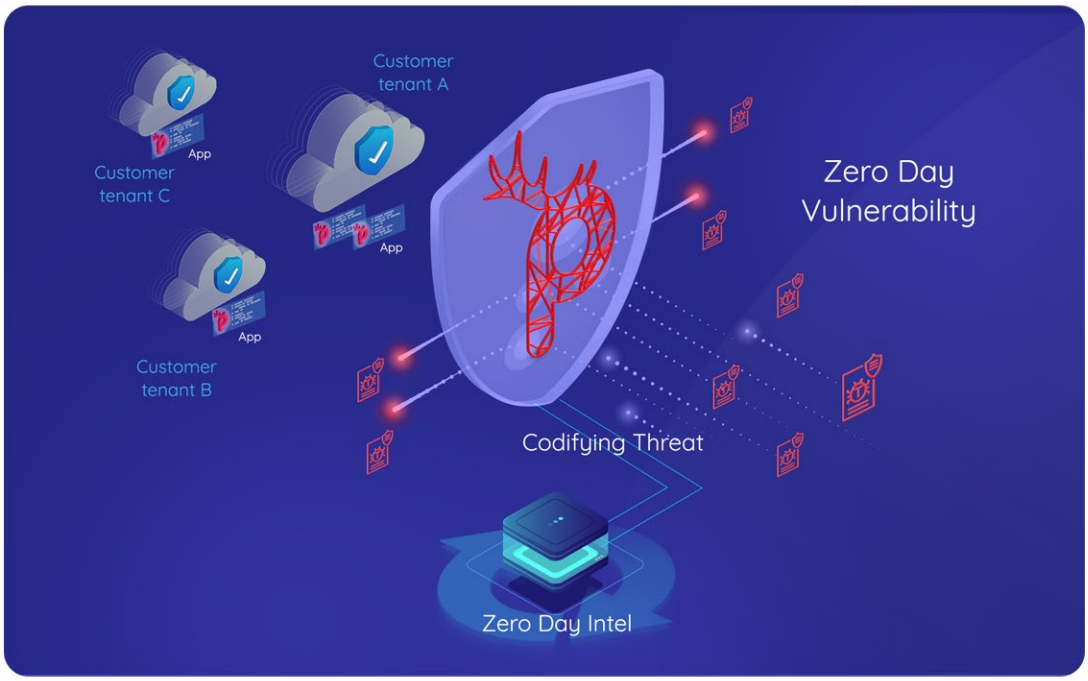
PAC enables pentesting in the developers' hands early in the SDLC, reflecting shift-left ideals. The Prancer CLI links seamlessly with CI pipelines to automatically check applications at each build as well as real-time testing in post-deployment pipelines.

The combination of these technologies creates a hybrid approach that combines the best elements of defense-in-depth and validation-in-depth security measures, allowing security teams to ensure attack-ready cloud applications are built.



09 Zeroday as a service (ZDaaS) for contextual security testing

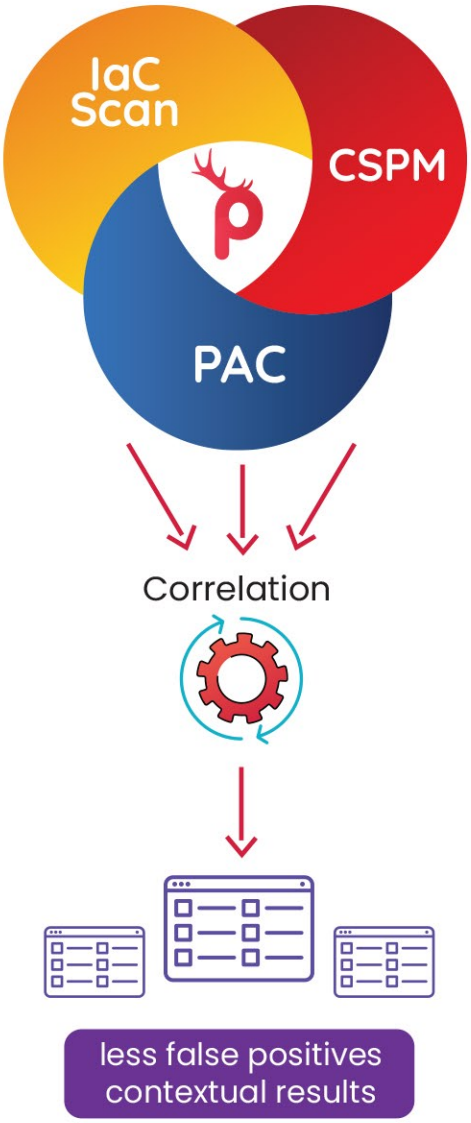
Zero-day vulnerabilities are on the rise, and being prepared for them is critical to avoid time-consuming firefighting; to combat zero-days, conventional protection methods must be tightly integrated with DevSecOps, which begins by recognizing vulnerabilities, rolling out both infrastructure and applications fixes, and contextual retesting end to end across the cloud apps at a scale



With Prancer's ZeroDay As A Service (ZDaaS), the PAC automatically discovers and pentests cloud-based applications, APIs, and services in scheduled or CI/CD cycles against zero-days and common cloud CVEs. In the most recent Log4J episode, Prancer was able to quickly identify and test all of the exposed APIs and web applications to get a comprehensive perspective on the exposure and confirm both cloud infrastructure and supplychain protections that were implemented as soon as it's deployed.

10 Stop the game of false positives

When multiple cloud security solutions are in use, it's a time-consuming task for a security expert to go through numerous findings, and simulation results and link them together to de-identify the false positives and compensated risks while still focusing on the high-risk items. It's now time to hone in on reducing false positives. The Prancer's context-aware policy engine and ZDaaS includes built-in capabilities that automatically correlate the Prancer CSPM and IAC scan findings and third-party findings with real-time pentest outcomes, lowering false positives.



11 Future of offensive security in the cloud

The future of offensive security in the cloud will be defined by APIs-driven, shift-lefted, DevOps compatible pentesting methods and techniques. Because modern cloud infrastructure and applications are based on APIs, it's conceivable for PAC engineers to come up with an automated pentesting framework that codifies the bespoke payloads and fuzzing approaches to check real-world critical vulnerabilities as they emerge in the wild.

12 Summary

Organizational security strategies must and will shift towards an end-to-end approach to security, rather than a siloed MPT/DAST approach. Automated penetration testing will develop organically, providing real value. It's a great match for organizations that want to secure cloud apps while saving time and scaling security consistently in an ever-changing threat landscape.

prancer

Cloud Security Validation at Scale